

Examiner's Amendment

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Elliot Chen on 7/15/11.

27.(Currently Amended) A method of organizing a set of application program interface (API) functions of an API layer in a program development computer system into a hierarchical namespace, the method comprising:

creating a plurality of groups in a file system that resides on a server of the program development computer system for the set of API functions according to type, each group containing logically related API functions of the application program interface; wherein one of the API functions communicates in first form and another of the API functions communicates in a second form that is incompatible with the first form;

assigning a name to each group using the program development computer system, wherein one of the groups includes at least one API function related to core concepts of the file system, wherein another of the groups includes at least one API function related to entities that a human being can contact, wherein another of the groups includes at least one API function related to document types that can be stored

in the file system, and wherein another of the groups includes at least one API function related to multiple kinds of media;

selecting a top level identifier and prefixing the name of each group with the top level identifier using the program development computer system so that the types in each group are referenced by a hierarchical name that includes the selected top level identifier prefixed to the name of the group containing the type; converting a communication associated with an API function of the API layer from the first form to the second form.

33(Currently Amended) A method for organizing functions in a program development computer system, the method comprising:

creating a first namespace in a file system that resides on a server of the program development computer system, the first namespace includes application program interface (API) functions of an API layer that enable identification of particular physical locations using the program development computer system, the API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program; the file system is included in a programming interface layer of the program development computer framework;

creating a second namespace on the file system that includes API functions of the API layer that enable identification of entities that can be contacted by a human being using the program development computer system, one of the API functions in the API layer communicates in a first form and another of the API functions in the API layer communicates in a second form that is incompatible with the first form; and converting a

communication associated with an API function of the API layer from the first form to the second form; wherein the file system is included in a programming interface layer of the program development computer framework.

42(Currently Amended) A method comprising:

calling one or more first application program interface (API) functions of an API layer that is running on top of a common language runtime layer that resides on a server of a program development computer system, the one or more first API functions enable documents to be described;

calling one or more second API functions of the API layer that are core functions expected to be used by the one or more first functions as well as a plurality of additional functions[.], and converting a communication associated with an API function of the API layer from a first form to a second form, wherein one of multiple API functions in the API layer communicates in the first form and another of the multiple API functions in the API layer communicates in the second form that is incompatible with the first form, and

wherein the one or more first functions, the one or more second functions, and the plurality of additional functions are organize in a file system in the program development computer system, the file system being included in the API.

47(Currently Amended) A method, comprising:

receiving one or more calls to one or more first application program interface (API) functions of the API layer that is running on top of a common language runtime

layer, the one or more first API functions enable identification of entities that can be contacted by a human being; ~~and~~

receiving one or more calls to one or more second API functions of the API layer that are core functions expected to be used by the one or more first API functions of the API layer as well as a plurality of additional API functions of the API layer using the program development computer system[[,]]; and converting a communication associated with an API function of the API layer from a first form to a second form, wherein one of multiple API functions in the API layer communicates in the first form and another of the multiple API functions in the API layer communicates in the second form that is incompatible with the first form, and

wherein the one or more first API functions of the API layer, the one or more second API functions of the API layer, and the plurality of additional API functions of the API layer are defined to organize a file system in a server of the program development computer system, the file system being included in a programming interface.

Reasons for Allowance

1. The following is an examiner's statement of reasons for allowance: listed below:

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

2. ***Status of Application:*** A Final action was mailed on 3/9/11, in which claims 1-21, 38-41, 50-54 were rejected under 35 U.S.C. 101, no art rejection was applied to the claims. In the Final action mailed 3/9/11, the Examiner indicated independent claims 1, 22, 27, 33, 38, 42, 47, and 50 were allowable.
3. Claims 1-54 are allowable.
4. Prior art fails to disclose or suggest, Claims 1, 22, 38, 50, "wherein one of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "providing a common language runtime layer that hands calls to the API functions of the program interface layer by the plurality of intermediate language programs to the operation system or the object model server for execution", and "converting a communication associated with an API function of the program interface layer from the first form to the second form", an example of prior art that fails to disclose or suggest these limitations is Marcey(7,305,677). Marcey discloses runtime based applications are transferred to dynamic runtime implementations suited to process the workload associated with the application code. Marcey discloses the common language is compiled to be

processed on a given platform. Marcey discloses that the application code is compiled into the CLI's common intermediate language and executed on the Common Language Infrastructure.

5. Prior art fails to disclose or suggest, Claims 27-32 "creating a plurality of groups in a file system that resides on a server of the program development computer system for the set of API functions according to type, each group containing logically related API functions of the application program interface", "wherein one of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "converting a communication associated with an API function of the API layer from the first form to the second form", an example of prior art that fails to disclose or suggest these limitations is Lennon(2002/0152267). Lennon discloses the grouping of "multimedia items", such as video, audio, and images, using XML schemas. Lennon discloses multimedia items conform to a schema, and that schemas are expressed or represented using the W3C schema language, XML Schema. Individual descriptions are represented using XML document instances. XML Schemas are also represented as XML documents. However, the grouping of "multimedia items" (e.g., video, audio, images) is not equivalent to the group of "API functions". Lennon fails to disclose or suggest, "creating a plurality of groups in a file system that resides on a server of the program development computer system for the set of API functions according to type, each group containing logically related API functions of the application program interface" ;"wherein one of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "converting a communication associated with an API function of the API layer from the first form to the

second form".

6. Prior art fails to disclose or suggest, Claims 33-37 "API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program", "wherein one of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "converting a communication associated with an API function of the API layer from the first form to the second form", and example of prior art that fails to disclose or suggest these limitations is Blackwell(2002/0183044). Blackwell discloses namespaces, such as a shell namespace. Blackwell discloses the shell namespace is referred to a messaging component, that may manage and unify user's message through Explorer shell interface. Blackwell discloses integration of the messaging component with the Explorer shell provides message management using namespace in addition to the components that come with the Windows, and allows the user to view additional data, such as a fax message, voice mail message, page messages, e-mail messages, or contact data. Blackwell fails to disclose or suggest, an "API interface layer running on top of a common language runtime layer to receive API function calls from an intermediate language program"; "wherein one of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "converting a communication associated with an API function of the API layer from the first form to the second form".

7. Prior art fails to disclose or suggest, Claims 42-49 "calling one or more first application program interface(API) functions of an API layer that is running on top of a common language runtime layer that resides on a server of a program development computer system"; "wherein one

of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "converting a communication associated with an API function of the API layer from the first form to the second form", an example of prior art that fails to disclose or suggest these limitation is, Mellmer(6,446,253). Mellmer discloses a hierarchically ordered plurality of namespaces, each namespace associated with at least one group of resources available to a program, each namespace comprising a plurality of names, each one of the plurality of names providing a logical interface to one of the resources in the at least one group wherein each name is a data structure including a pointer to an object associated with the one of the resources in the at least one group and a mapping to a routine for implementing the retrieval of the associated object from an abstract storage system. Mellmer fails to disclose or suggest, "calling one or more first application program interface (API) functions of an API layer that is running on top of a common language runtime layer that resides on a server of a program development computer system"; "wherein one of the API functions communicates in a first form and another of the API functions communicates in a second form that is incompatible with the first form"; "converting a communication associated with an API function of the API layer from the first form to the second form".

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JENISE JACKSON whose telephone number is (571)272-3791.

The examiner can normally be reached on Increased flex time, work at home every Monday,
Tues-Fri office hours 8:00-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Edan Orgad can be reached on (571) 272-7884. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

July 15, 2011
/J. J./
Examiner, Art Unit 2439

/Edan Orgad/
Supervisory Patent Examiner, Art Unit 2439